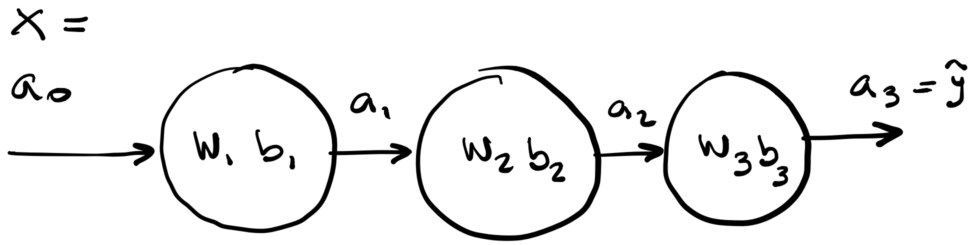
ENTRENAMIENTO DE UNA RED NEURONAL:



- Ecuaciones para generar $\hat{y}$ (FORWARD PROPAGATION)

$$z_k = W_k a_{k-1} + b_k$$

$$a_k = \sigma(z_k) \qquad k = 1, 2, \dots m = 3$$

$$\hookrightarrow \sigma(z) = \frac{1}{1+e^{-z}}$$

$$* \quad \sigma'(z_k) = \dots = a_k(1-a_k)$$

- Método de Gradiente:

(W_k, b_k) para $k=1,\dots m$ se estiman minimizando una función objetivo, por ejemplo:

$$J(\theta) = \frac{1}{N} \sum \frac{1}{2} (\hat{y}_i - y_i)^2$$

$\theta = [W_1, b_1, W_2, b_2 \dots W_m, b_m]$
para N muestras de entrenamiento (x_i, y_i).

$W_k, b_k \leftarrow$ se inician con valores aleatorios

$$\left.\begin{aligned} W_k &:= W_k - \alpha \frac{\partial J}{\partial W_k} \\ b_k &:= b_k - \alpha \frac{\partial J}{\partial b_k} \end{aligned}\right\}$$

W_k, b_k se estiman de forma iterativa $\longrightarrow$ hasta obtener convergencia.

- Derivadas Parciales:

$$\Delta W_k = \frac{\partial J}{\partial W_k} = \underbrace{\frac{\partial J}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k}}_{\delta_k} \cdot \overbrace{\frac{\partial z_k}{\partial w_k}}^{a_{k-1}} = \delta_k \, a_{k-1}$$

$$\Delta b_k = \frac{\partial J}{\partial b_k} = \frac{\partial J}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k} \cdot \underbrace{\frac{\partial z_k}{\partial b_k}}_{1} = \delta_k$$

Donde

$$\delta_k = \frac{\partial J}{\partial a_k} \cdot \overbrace{\frac{\partial a_k}{\partial z_k}}^{\text{ver } (*)} = \frac{\partial J}{\partial a_k} \cdot a_k(1-a_k)$$

$\frac{\partial J}{\partial a_k}$ se encuentra de atrás para adelante:

$$\longrightarrow \quad \frac{\partial J}{\partial a_m}, \frac{\partial J}{\partial a_{m-1}}, \dots \frac{\partial J}{\partial a_k}, \frac{\partial J}{\partial a_{k-1}} \dots \frac{\partial J}{\partial a_0}$$

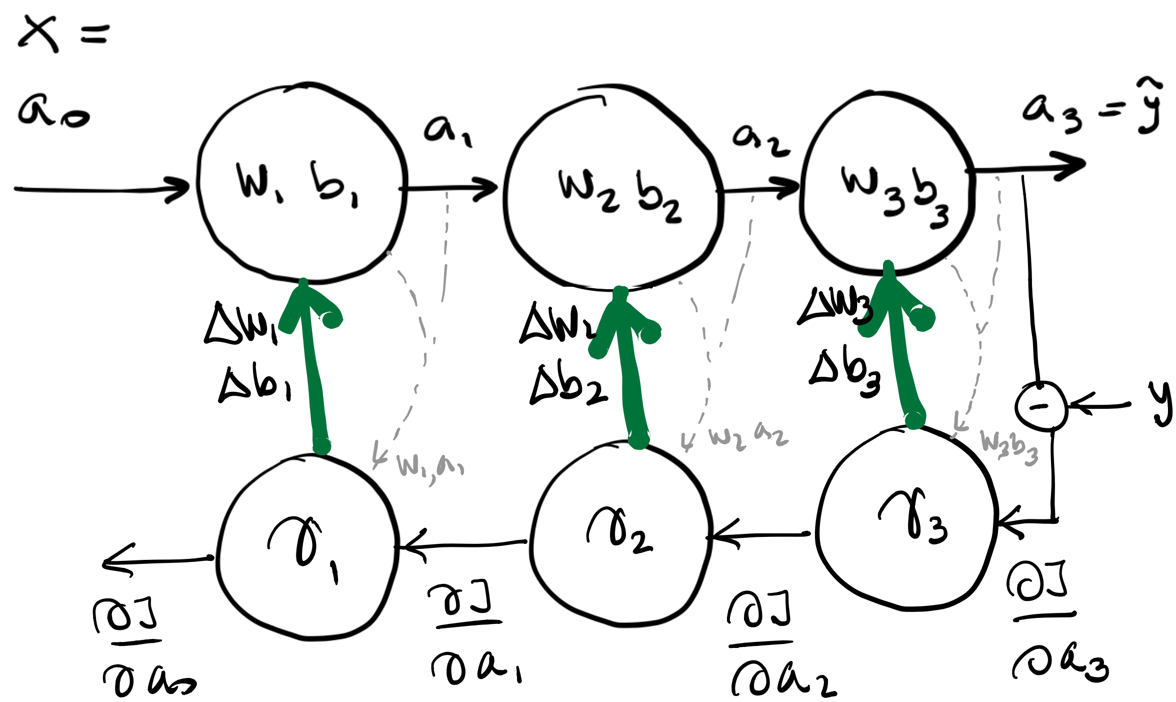
donde

$$\frac{\partial J}{\partial a_m} = \frac{\partial J}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}} \left\{ \frac{1}{N} \sum_i \frac{1}{2}(\hat{y}_i - y_i)^2 \right\} = \frac{1}{N} \sum_i (\hat{y}_i - y_i)$$

$\vdots$

$$\frac{\partial J}{\partial a_{k-1}} = \underbrace{\frac{\partial J}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_k}}_{\delta_k} \cdot \overbrace{\frac{\partial z_k}{\partial a_{k-1}}}^{W_k} = \delta_k W_k$$

$\vdots$



ALGORITMO:

1. INICIALIZACION RANDOM:

$$(W_k, b_k)_{k=1 \ldots m} = \text{random}$$

2. FOREWARD PROPAGATION:

$$z_k = W_k a_{k-1} + b_k$$

$$a_k = \sigma(z_k)$$

3. DERIVADAS PARCIALES:

$$(\Delta W_k, \Delta b_k)_{k=1 \ldots m} = \text{zeros}$$

for $k = m, m-1, \dots 0$

if $k = m$?

$$\Delta a_k = \hat{y} - y$$

$$\frac{\partial a_k}{\partial z_k} = a_k(1 - a_k)$$

$$\delta_k = \Delta a_k \cdot \frac{\partial a_k}{\partial z_k}$$

$$\Delta W_k = \delta_k a_{k-1}$$

$$\Delta b_k = \delta_k$$

$$\Delta a_k = \delta_k \cdot W_k$$ (usado en la proxima como $k-1$)

end

4. ACTUALIZACION DE W_k, b_k

$$\left.\begin{aligned} b_k &= b_k - \alpha \Delta b_k \\ W_k &= W_k - \alpha W_k \end{aligned}\right\} \text{ para } k = 1 \dots m$$

5. CRITERIO DE CONVERGENCIA

$$J = \| y - \hat{y} \| > \varepsilon \text{ AND } \# iter > iter\_max$$

$\rightarrow$ GOTO 2

$\rightarrow$ ELSE END